Postel

Network Working Group
Request for Comments: 1141
Obsoletes: RFC 1071

T. Mallory
A. Kullberg
BBN Communications
January 1990

Incremental Updating of the Internet Checksum

Status of this Memo

This memo correctly describes the incremental update procedure for use with the standard Internet checksum. It is intended to replace the description of Incremental Update in RFC 1071. This is not a standard but rather, an implementation technique. Distribution of this memo is unlimited.

Description

In RFC 1071 on pages 4 and 5, there is a description of a method to update the IP checksum in the IP header without having to completely recompute the checksum. In particular, the RFC recommends the following equation for computing the update checksum C' from the original checksum C, and the old and new values of byte m:

$$C' = C + (-m) + m' = C + (m' - m)$$

While the equation above is correct, it is not very useful for incremental updates since the equation above updates the checksum C, rather than the 1's complement of the checksum, ~C, which is the value stored in the checksum field. In addition, it suffers because the notation does not clearly specify that all arithmetic, including the unary negation, must be performed one's complement, and so is difficult to use to build working code. The useful calculation for 2's complement machines is:

$$\sim C' = \sim(C + (-m) + m') = \sim C + (m - m') = \sim C + m + \sim m'$$

In the oft-mentioned case of updating the IP TTL field, subtracting one from the TTL means ADDING 1 or 256 as appropriate to the checksum field in the packet, using one's complement addition. One big-endian non-portable implementation in C looks like:

```
unsigned long sum;
ipptr->ttl--;                          /* decrement ttl */
sum = ipptr->Checksum + 0x100;  /* increment checksum high byte*/
ipptr->Checksum = (sum + (sum>>16)) /* add carry */
```

This special case can be optimized in many ways: for instance, you

can bundle updating and checking the ttl. Compiler mileage may vary. Here is a more general and possibly more helpful example which updates the ttl by n seconds:

```
UpdateTTL(iph,n)
struct ip_hdr *ipptr;
unsigned char n;
{
    unsigned long sum;
    unsigned short old;

    old = ntohs(*(unsigned short *)&ipptr->ttl);
    ipptr->ttl -= n;
    sum = old + (~ntohs(*(unsigned short *)&ipptr->ttl) & 0xffff);
    sum += ntohs(ipptr->Checksum);
    sum = (sum & 0xffff) + (sum>>16);
    ipptr->Checksum = htons(sum + (sum>>16));
}
```

Security Considerations

Security issues are not addressed in this memo.

Authors' Addresses

Tracy Mallory
BBN Communications Corporation
50 Moulton Street
Cambridge, MA 02238

Phone: (617) 873-3193

EMail: tmallory@CCV.BBN.COM

A. Kullberg
BBN Communications Corporation
50 Moulton Street
Cambridge, MA 02238

Phone: (617) 873-4000

EMail: akullberg@BBN.COM